



Towards Real-Time Computation of Cardiac Electrophysiology for Training Simulator

Hugo Talbot, Christian Duriez, Hadrien Courtecuisse, Jatin Relan, Maxime Sermesant, Stéphane Cotin, Hervé Delingette

► To cite this version:

Hugo Talbot, Christian Duriez, Hadrien Courtecuisse, Jatin Relan, Maxime Sermesant, et al.. Towards Real-Time Computation of Cardiac Electrophysiology for Training Simulator. Statistical Atlases and Computational Models of the Heart - STACOM 2012 in the 15th International Conference on Medical Image Computing and Computer Assisted Intervention - MICCAI 2012, Oct 2012, Nice, France. hal-00750835

HAL Id: hal-00750835

<https://inria.hal.science/hal-00750835>

Submitted on 12 Nov 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Towards Real-Time Computation of Cardiac Electrophysiology for Training Simulator

Hugo Talbot^{1,2}, Christian Duriez², Hadrien Courtecuisse², Jatin Relan¹,
Maxime Sermesant¹, Stéphane Cotin², Hervé Delingette¹

¹ ASCLEPIOS Group, INRIA Sophia-Antipolis, France

² SHACRA Group, INRIA Lille North Europe, France

Abstract. This work aims at developing a training simulator for interventional radiology and thermo-ablation of cardiac arrhythmias. To achieve this, a real-time model of the cardiac electrophysiology is needed, which is very challenging due to the stiff equations involved. In this paper, we detail our contributions in order to obtain efficient cardiac electrophysiology simulations. First, an adaptive parametrisation of the Mitchell-Schaeffer model as well as numerical optimizations are proposed. An accurate computation of both conduction velocity and action potential is ensured, even with relatively coarse meshes. Second, a GPU implementation of the electrophysiology was realised in order to decrease the computation time. We evaluate our results by comparison with an accurate reference simulation using model parameters, personalized on patient data. We demonstrate that a fast simulation (close to real-time) can be obtained while keeping a precise description of the phenomena.

1 Introduction

Cardiac arrhythmias are characterised by a pathological electrical activity in the myocardium (heart muscle) which can be lethal. Catheter thermo-ablation is a minimally invasive technique that can prevent fibrillation by removing the substrate or the trigger of such a phenomenon through the ablation of the responsible cardiac cells. This procedure is performed by highly skilled and experienced interventional cardiologists, yet the success rate remains limited due to the complexity of such pathologies. Moreover, due to population ageing, an important increase is predicted in such arrhythmias, without an associated increase in interventional cardiologists. Therefore, there will be an important need for novel tools in learning, training and planning, such as simulators.

One important missing step towards the development of such a simulator is the ability to simulate the electrophysiology of the heart in real-time. The electrical wave propagating inside the cardiac walls corresponds to ion exchanges between the cells and through the cell membrane. The myocardial cells are polarised, i.e. there is a potential difference between the inside and the outside of the cells. This potential difference is called action potential (AP) or transmembrane potential. Before a thermo-ablation, cardiologists first carry out a diagnostic electrophysiological study (programmed electrical stimulation) allowing a

definitive diagnosis. From the acquired data, the ablation intervention can be planned. We focus here on the training of cardiologists for these catheter-based ablation of cardiac tissues in the context of cardiac arrhythmias.

Research on cardiac electrophysiology models is very active. The proposed models can be divided into three different classes: (i) biophysical models, which are complex models including the different ionic concentrations and channels, involving many parameters and simulating the electrophysiology close to the cellular scale [14]; (ii) phenomenological models, which are simplified models [13], [1], [7], [10] derived from the biophysical models, involving less parameters and capturing the AP shape and its propagation at the organ scale; (iii) Eikonal models [8], which correspond to static non-linear partial differential equations for the depolarization time derived from the previous models. These models enable to simulate wave propagation but cannot accurately account for complex physiological states and the parameters have no direct physiological meaning.

This paper addresses the problem of real-time simulation of electrophysiology thanks to a new modelling approach adapted to efficient computational strategies. The approach relies on one of the phenomenological models from Mitchell-Schaeffer (MS) [10]. The following sections presents the MS model, the model parameters and the numerical settings. Then we detail how we adapt the model parameters and numerical settings in order to recover patient-specific features, namely the conduction velocity (CV) and the action potential duration (APD, denoting APD90). To further improve our computation times, we propose an efficient GPU implementation with CUDA. Finally, we present our first results on patient-specific data and we conclude with some ideas for future work.

2 Real-Time Cardiac Electrophysiology Simulation

2.1 Discretisation of the Cardiac Electrophysiology Model

Mitchell-Schaeffer Model. For the last fifty years, numerous models were proposed for the cardiac cell AP. Regarding the features of the model categories presented previously, the phenomenological models meet best our requirements in terms of computational efficiency at the organ scale. The model that we chose is the MS model [10] because of the following reasons: (i) it has only 5 parameters that we detail below, (ii) each parameter has a physiological meaning and (iii) it provides a better estimation of the AP compared to other phenomenological models (as the Aliev-Panfilov model [1]).

The MS model is a two-variable model derived from the Fenton Karma model [7]. The equations describing the model are written in 1:

$$\begin{cases} \partial_t u = \text{div}(D\nabla u) + \frac{zu^2(1-u)}{\tau_{in}} - \frac{u}{\tau_{out}} + J_{stim}(t) \\ \partial_t z = \begin{cases} \frac{(1-z)}{\tau_{open}} & \text{if } u < u_{gate} \\ \frac{-z}{\tau_{close}} & \text{if } u > u_{gate} \end{cases} \end{cases} \quad (1)$$

where u is a normalized transmembrane potential¹ and z is the gating variable associated to the sodium ion influx, thus depicting the repolarization phase. The extra-cellular potential can be recovered using the method described in [4]. The diffusion term is defined by an 3x3 anisotropic diffusion tensor $D = d \cdot \text{diag}(1, r, r)$ so that the planar conduction velocity in the fiber direction is 2.5 times greater than in the transverse plane ($r = \frac{1}{(2.5)^2}$). d is the diffusion coefficient. The parameters τ_{in} and τ_{out} define the repolarization phase whereas the constants τ_{open} and τ_{close} manage the gate opening or closing depending on the change-over voltage u_{gate} . The term $J_{stim}(t)$ is the stimulation current applied in the pacing area. The default values (describing the common action potential) of these parameters are given in [10]. The initial conditions of our simulation were set as Dirichlet conditions. The pacing area ($u_0 = 1$ and $z_0 = 1$) has been extracted from the depolarization time map.

Patient-Specific Spatial Discretisation. The cardiac geometry that we used in this study is an anatomy segmented from a patient MRI. This patient suffers from chronic ischemia and a left bundle branch block (noted LBBB).

The method used for the spatial discretization of the model is the finite element method which requires a volumetric mesh of the myocardium. The volumetric bi-ventricular geometry has been meshed with linear tetrahedra using CGAL (www.cgal.org) algorithms. The edge size used is about $dx = 4 \text{ mm}$ and implies 65547 tetrahedra which is already substantial for real-time simulation. Usually the usual edge size required for cardiac electrophysiology is $dx < 0.1 \text{ mm}$. However the element size has to be fine enough to correctly model the wave propagation, especially in the grey zones around scars. Scar regions are known to be very unstable electrical zones and can be responsible for cardiac reentries. These areas therefore need to be discretized with accuracy.

A LBBB abnormal conduction implies a late activation of the left ventricle. Regarding this pathology, the edge size of 4 mm should be sufficient for our simulation since only the times of depolarization and repolarization have to be computed accurately. However to model a possible ventricular tachycardia induced by an isthmus the edge size should be finer. As demonstrated in [9], a 0.8 mm edge size must be used for accurate simulation.

Time Integration Scheme. For all our simulations, we use the second-order semi-implicit solver called Modified Crank Nicholson/Adams Bashforth (MC-NAB), detailed in [6]. This solver defines an implicit integration of the diffusion term and an explicit integration of the ionic current. The construction of the matrix system leads to a symmetric definite positive case that we can solve using many existing algorithms.

In our GPU simulations, we work with a conjugate gradient (CG) to solve the resulting linear system (written $\mathbf{A}x = b$). To improve the convergence rate, we

¹ The MS is a monodomain model since it is expressed according to the transmembrane potential whereas bidomain models depends on both intra- and extra-cellular potential.

use a preconditioner Jacobi. The Jacobi method computes the inverse matrix of the diagonal of the system matrix \mathbf{A} . Since our matrix \mathbf{A} is diagonal dominant, the use of this preconditioner is straightforward. The factorization provided by the Jacobi method can be updated when needed, thus allowing to interact with the model (e.g. thermo-ablation).

2.2 Numerical Study

To achieve our performance goal, the numerical settings of our simulation need to be optimized. In the literature about cardiac electrophysiology simulation, it is shown that both element size and time step have to be small enough to capture the cardiac phenomenon. Usual values for the edge size are defined such that $dx \leq 0.1 \text{ mm}$ and the time step is lower than $dt \leq 0.01 \text{ ms}$ (see [11]). However meshing a heart geometry with such small elements would imply millions of tetrahedra. Such small elements as well as very small time steps prevent real-time simulations. To achieve real-time performances, we have to use larger time steps and larger finite-element. In this part, we study the impact of using large elements, the limitation in terms of time step as well as solutions to reach real-time performances.

Locally-Adapted Model Parameters To ensure the reliability of the training simulation, we need a relatively accurate computation of the CV. Nevertheless using large elements will affect the diffusion effect, i.e. the wave propagation and its CV. We studied the influence of the element size on the CV with the lumping using the MS model on a 3D bar ($15\text{cm} \times 2.5\text{cm} \times 2.5\text{cm}$, regularly meshed) on which the wave is propagating along the bar axis with a planar front. A similar study was conducted in [11] where they also considered other integration method for the ionic term than the lumping method. The results are compared to a very accurate computation of the MS model in 1D using Matlab with a time step $dt = 10^{-5} \text{ s}$ and a spatial step $dx = 5 \cdot 10^{-6} \text{ m}$ that provides a CV reference value: $CV_{ref} = 0.5124 \text{ m/s}$. In the Fig. 1, the results show that the CV decreases when the mesh becomes coarse.

A way to compensate this integration error is to artificially increase the diffusion coefficient d . Based on this idea, a personalization of the diffusion coefficient d and the opening and closing time constants τ_{open} and τ_{close} has been computed to fit measured patient data, using the personalization method of [12]. This personalization step gave us values of the diffusion coefficient for each tetrahedron, as well as values of opening and closing time constants on each vertex.

Time-Step Limitations Even if the time step does not affect the electrophysiology features, the explicit formulation of the ionic term implies a stability condition on the time step. This limitation of the time step has already been pointed out in [6] and [5]. Since the diffusion is computed implicitly the limiting term is the ionic term. Following [5], we should define $dt \leq \frac{1}{\inf(\partial f)} = \frac{\tau_{in}\tau_{out}}{\tau_{in} + \tau_{out}} = 0.286 \text{ ms}$, if we would use a semi-implicit Euler solver (f denoting the ionic term).

Using the MCNAB, we measured $dt \leq 0.59 \text{ ms}$. For more stability, we will use $dt = 0.4 \text{ ms}$.

Optimization on CG tolerance As detailed in 2.1, we use a CG to solve our system. Our solver stops when the difference between the results given by two successive iterations is below the tolerance, noted tol . The tolerance $tol_{ref} = 10^{-10}$ ensures us to reach the "exact" solution. However to improve the computation performances, we computed the optimal tolerance value. We found the tolerance $tol_{opt} = 10^{-6}$ giving a L^2 -error about $err = 3.79 \cdot 10^{-5}$ on the action potential field, which is fully acceptable. In the results part, we will show performances using both tolerances $tol_{ref} = 10^{-10}$ and $tol_{opt} = 10^{-6}$ in section 3.

2.3 GPU Implementation

The GPU architecture consists in several multi-processors able to carry out highly parallel tasks independently. The complexity of GPU programming results in defining an optimal distribution of the threads and minimizing the memory access latency. In our approach, we relied on the CUDA toolkit (dedicated to NVidia's GPUs) to develop the GPU version of the method. Similar results could be obtained using OpenCL and other GPU models.

The implementation of the ionic term of MS is based on classical parallelization methods. Using the lumping integration method, the ionic term is computed on each vertex. Therefore each thread is dedicated to one vertex and computes the contribution of the MS term for this vertex. However our computation strategy ensures a tiled access in memory in our GPU code.

The diffusion term $div(D \cdot grad(u))$ is more complicated to implement in parallel. The contribution of the diffusion term is computed from the edges and summed on each vertex. In a parallel computation, this algorithm can lead to writing conflicts: two threads solving two adjacent edges could write on the same point simultaneously. New GPUs supporting CUDA 2.0 handle now these

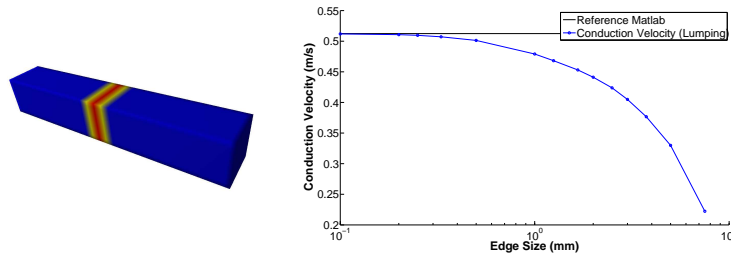


Fig. 1. (Left) initial conditions on the 3D bar (Right) Evolution of the CV (m/s) against element size (mm).

atomic operations. Nevertheless a solution to this problem is detailed in [2] which is twice faster than atomic functions from CUDA. We decided to adapt this algorithm, originally designed for deformable finite element equations, to our diffusion model. This technique consists in first computing the neighborhood of each node. Then the contribution of the diffusion is computed and saved on all the edges. Finally, the contribution on each vertex is obtained by accumulating the values computed on the edges with the neighborhood information. In this last accumulation step, a parallel reduction is carried out using several threads by vertex as explained in [2].

Our GPU implementation also includes a conjugate gradient and a Jacobi preconditioner that are used to solve very quickly the electrophysiology model on the whole mesh. Only single float precision has been considered since it is sufficient regarding the electrophysiology accuracy. Moreover simulations using single float precision are about two times faster than simulations using double precision as detailed in [3].

Finally, it must be stressed that this GPU implementation is a novel application of an efficient method originally proposed by [2] for deformable finite element computations. This parallel implementation will be crucial in our performance results.

3 Results and Discussion

The local adaptations on the MS model that we detailed in 2.2 will still generate some errors. We now want to evaluate this error in comparison with our ground-truth data. Moreover a GPU implementation has been done to improve the computation time of our simulation. The performances using the optimized model parameters will also be assessed with this implementation.

3.1 Depolarization Times and APD Error Maps

We compared our results with the reference simulation using very small time step ($dt = 10^{-5}$ s) and model parameters fitting patient data. We computed an average error on depolarization time of 4.05 ms ($\sim 5.48\%$) and an average error on APD of 19.1 ms ($\sim 5.62\%$). These results shows that the errors are bounded. As future work, the restitution analysis of this patient-specific case seems to be compulsory to fully validate our simulation.

3.2 GPU Performances

Using the optimization presented previously would not be sufficient for fast computations. However, when combined to a GPU implementation, we obtain a significant speedup. Table 3.2 sums up computation times and other metrics of performance on the CPU and on the GPU (GeForce GTX580 with 512 cores). The processor used for CPU computations is a Intel Xeon W3550.

	CPU	GPU	GPU (using tol_{opt})
Mean computation time for one time step (ms)	25.4	3.89	2.17
real-time ratio (slower than real-time)	75.8	7.04	2.95

Table 1. Performance comparison between CPU and GPU.

From this table, we clearly notice the performance gain offered by the parallel implementation. The GPU simulation is more than 10 times faster than the CPU one. The computation time spent in the computation of diffusion and MS becomes negligible. Now the limitation of our simulation is now the solver, since it is the most time-consuming part of our computation. We notice that reasonably decreasing the tolerance of the CG significantly improves the performances.

To discuss these results, it can be stressed that 65547 tetrahedra represent a large amount of computation in a real-time context. A first solution would be to use on a coarser mesh. It must be stressed that the electrical wave of this patient was especially fast implying a small time step. Next generation GPU cards could also be most helpful.

3.3 User Interaction

A simulation dedicated to a training system assumes that cardiologists will interact with it. Our simulations already allow to stimulate any area of the heart in real-time (constraint with $u = z = 1$). This allows to simulate the stimulation done by the surgeon before the ablation in order to set the definitive diagnosis.

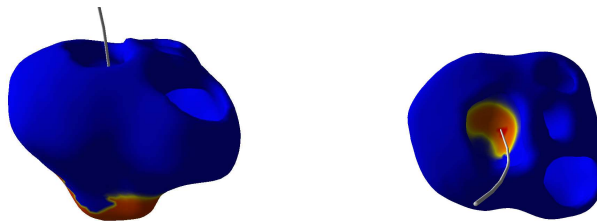


Fig. 2. Two different views of an electrical stimulation (with a catheter) applied by the user on a two-chamber geometry (ventricles).

The procedure of thermo-ablation is also handled by our computations. The cardiac cells treated by thermo-ablation can not conduct the current anymore. These cells can be seen as zero conductivity area. To take this change into account, we update the factorization computed by the Jacobi method. This thermo-ablation step can therefore be done without affecting the computation times.

4 Conclusion

In the coming years, the growing number of cardiology students implies to improve the training methods. Medical training simulators are an efficient solution to answer this need. Their main advantage is that simulators enable to train on virtual patient, thus avoiding to early operate on patients. Moreover such simulators could be a way of improving the course of medical studies.

In this paper, we presented some contributions to speed up simulations of cardiac electrophysiology. First, we locally modified time constant parameters from the Mitchell-Schaeffer model in order to use larger time steps and a coarser mesh. Then, we calculated the optimal time step and tolerance value for our iterative solver. Finally, we implemented a GPU version of the code modeling the diffusion and the Mitchell-Schaeffer model. With these optimizations, we reached an interactive simulation close to real-time based on 3D patient-specific data. This work is a very encouraging step toward the development of a training simulator.

In this work, only monocycle simulations have been considered, i.e. simulating only one cardiac cycle. A multicycle study needs to be carried out to analyze the restitution curves of our model and to validate our simulations. This paper presents a personalized simulation using data from a patient suffering from LBBB.

In the future, we want to address the challenging issue of simulating the electrophysiology of patients suffering from a ventricular tachycardia with an isthmus. This would lead to compute models with finer elements around the regions of interest (scars and isthmus) in order to accurately capture the electrical activity around these regions. To do so, we plan to implement innovative numerical method to use finer meshes without altering our performances.

References

1. Aliev, R.R., Panfilov, A.V.: A simple two-variable model of cardiac excitation. *Chaos, Solitons and Fractals* 7(3), p. 293–301 (1996)
2. Allard, J., Courtecuisse, H., Faure, F.: Implicit fem solver on gpu for interactive deformation simulation. In: *GPU Computing Gems Vol. 2*. NVIDIA Elsevier (2011)
3. Bartocci, E., Cherry, E.M., Glimm, J., Grosu, R., Smolka, S.A., Fenton, F.H.: Toward real-time simulation of cardiac dynamics. In: *Proceedings of the 9th International Conference on CMSB*. pp. p. 103–112. ACM (2011)
4. Chhay, M., Coudière, Y., Turpault, R.: How to compute the extracellular potential in electrocardiology from an extended monodomain model. Research Report RR-7916, INRIA (2012)
5. Coudière, Y., Pierre, C.: Stability and convergence of a finite volume method for two systems of reaction-diffusion equations in electro-cardiology. *Nonlinear Analysis: Real World Applications* 7(4), p. 916–935 (2006)
6. Ethier, M., Bourgault, Y.: Semi-implicit time-discretization schemes for the bidomain model. *SIAM J. Numerical Analysis* 46, p. 2443–2468 (June 2008)
7. Fenton, F., Karma, A.: Vortex dynamics in three-dimensional continuous myocardium with fiber rotation. *Chaos* 8(1), p. 20–47 (1998)
8. Keener, J.P.: An eikonal-curvature equation for action potential propagation in myocardium. *Journal of Mathematical Biology* 29, p. 629–651 (1991)
9. Lamecker, H., Mansi, T., Relan, J., Billet, F., Sermesant, M., Ayache, N., Delingette, H.: Adaptive tetrahedral meshing for personalized cardiac simulations. In: *CI2BM09 - MICCAI Workshop on Cardiovascular Interventional Imaging and Biophysical Modelling*. pp. 149–158 (2009)
10. Mitchell, C., Schaeffer, D.: A two-current model for the dynamics of cardiac membrane. *Bulletin of Mathematical Biology* 65, p. 767–793 (2003)
11. Pathmanathan, P., M.G.S.J., Whiteley, J.: The significant effect of the choice of ionic current integration method in cardiac electro-physiological simulations. *International Journal for Numerical Methods in Biomedical Engineering* (2011)
12. Relan, J., Sermesant, M., Delingette, H., Pop, M., Wright, G., Ayache, N.: Quantitative comparison of two cardiac electrophysiology models using personalisation to optical and mr data. *IEEE International Symposium* pp. 1027–1030 (2009)
13. Richard, FitzHugh: Impulses and physiological states in theoretical models of nerve membrane. *Biophysical Journal* 1(6), p. 445–466 (1961)
14. Tusscher, K.H.W.J.T., Noble, D., Noble, P.J., Panfilov, A.V.: A model for human ventricular tissue. *American Journal of Physiology - Heart and Circulatory Physiology* 286(4), p. 1573–1589 (April 2004)